REMARKS

Applicant has considered the Office Action mailed on October 6, 2003, and the references cited therewith. This response cancels no claims; amends claims 1, 6, 9, 12, 16-18, and 25; and adds new claims 29-31. As a result, claims 1-31 are now pending in this Application.

Drawing Objection

The drawing was objected to for not showing reference numeral "700" described in the Specification. The attached replacement sheet shows this numeral in the place described for it in the text. Therefore, this correction introduces no new matter prohibited by 35 USC §132.

Claim Objections

Claims 17 and 18 were objected to due to informalities. Applicant believes that the amendments to claims 17 and 18 obviate these objections without introducing any new matter under 35 USC §132.

Claim Rejections

Claims 1, 4, 5 and 12-28 were rejected under 35 USC §103(a) as being unpatentable over Dundas et al. (Proceedings of the 1997 International Conference on Supercomputing, pp. 68-75 (1997)) in view of Ukai et al. (U.S. patent 5,983,324). Claims 2-3 and 6-11 were rejected under 35 USC § 103(a) as being unpatentable over Dundas et al. in view of Ukai et al., and further in view of Petrick et al. (U.S. 5,920,889).

The primary reference to Dundas teaches only one thing relevant to the present invention: the concept of run-ahead processing after a cache miss. Applicant finds no suggestion within Dundas that run-away run-ahead processing could step on its own feet and actually create a worse situation by evicting cache lines that needed in subsequent normal processing before the newly prefetched lines that evicted them. Dundas simply does not recognize that such a problem could arise.[1]

Ukai describes a system quite different from Dundas' high-speed cache associated with a single processor chip. Ukai concerns an OS-controlled main-storage buffer area for handling large, variable-size files requested by multiple user processes executing concurrently. "Complex" is too poor a word to describe Ukai's system. For example, certain kinds of "sequential accesses" (Fig. 3, block 45) determine whether stopping prefetching is even considered. Complicated algorithms determine how many files can reside in the buffer area at once. Whether files are shared among different user processes affects protection. Even so, his algorithms may allow a data item to be overwritten before the protection even starts (e.g., col. 4:46-51.[2]

Moreover, Ukai has no concept of operating in different modes at different times, one for normal processing and another mode for prefetching files. Ukai prefetches files continuously, stopping only when his dedicated main-storage buffer is full or (one assumes) when all the files requested by currently executing user processes have been fetched from a disk drive. Simultaneously, user processes read or write to files in the buffer area continuously, throughout the time that files are being fetched into the buffer area. That is, Ukai's prefetching and user processing of files operate entirely concurrently and asynchronously with regard to each other. Applicant, in contrast, employs run-ahead prefetching only when normal processing is interrupted by an event such as a cache miss, or in a separate thread in a multi-threaded processor. Applicant's protection method operates only during a run-ahead mode that is separate in time or thread from the normal execution of instructions.

The tertiary reference to Petrick is irrelevant, because it teaches no run-ahead at all.[3] The term "prefetch" in this patent refers only to a microinstruction that precedes the load and store of individual lines in his cache during normal operation of the processor; see, for example, col. 4:67-5:4 and col. 5:14-20. That is, Petrick has no time- or thread-separated modes during which he performs either run-ahead prefetching or normal instruction execution. And, of course,

[1] ---This is so even though Ukai's date precedes publication of Dundas' paper by almost a year and a half. This period corresponds to a doubling of processor size in the fast-moving field described in the well-known Moore's Law.

[2] ---The colon notation indicates column and line numbers. Thus, "col. 4:46-51" signifies "column 4, lines 46-51."

[3] ---Patrick's purpose is an improved implementation of a particular kind of small cache, a "copy-back" cache.

Petrick no protection associated with individual cache lines---nor would any such protection serve any purpose in his system.

Applicant respectfully traverses the rejection of claims 1-5 under 35 USC §103. Independent claim 1 determines whether a processor mode is "run-ahead execution or normal execution." Ukai both prefetches and uses files at all times, and thus has no such modes, nor any need for them. Although Dundas performs run-ahead and normal execution at different times, he has no teaching for---nor any purpose for--- "determining" which mode his system is in at any given time, because that fact would make no difference in the operation of his system. Although these references taken together thus does not reach claim 1, the very combination of them can be motivated only by an impermissible hindsight reconstruction of the claimed invention. That is, why would one skilled in the art be motivated to seek an answer to a problem that Dundas does not even raise?

Dependent claims 2-5 incorporate all the recitations of parent claim 1, and therefore also distinguish over Dundas and Ukai. As noted earlier, the Petrick reference does not concern run-ahead prefetching, and thus adds nothing to the other two references. And for that reason, one of ordinary skill would not be motivated to combine Petrick with the others

The rejection of claims 6-8 as unpatentable over Dundas in view of Ukai and further in view of Petrick is traversed for similar reasons. Independent claim 6 recites that the potential victim is found during a run-ahead mode that is "separate from a normal execution mode" of the processor. Ukai has no such separate modes, and Dundas does not perform any operations such as determining a protection-bit status or evicting a line only under a condition associated with such a bit. Again, Petrick has neither separate modes nor protection bits. Hence, the combination of these three references does not reach the claimed combination, and is not motivated by teachings within the references themselves, as required by 35 USC §103. Claims 7-8 depend from claim 6.

The rejection of claims 9-11 is traversed for reasons similar to those adduced in connection with claim 1, above. Ukai has no separate modes, and Dundas need not ever determine which mode his processor is in for any purpose. Because Dundas does not even recognize the overrun problem during run-ahead, he neither has nor needs any "protection bit" associated with a cache line. Thus, Dundas and Ukai do not attain to the language of independent claim 9, and there is no motivation for the combination in any case. Claims 10-11 depend from claim 9.

The rejection of claims 12-15 is respectfully traversed. Par. 5 of the Office Action treats independent claim 12 as equivalent to claim 1 under the cited art. However, claim 12 includes a number of recitations of separate threads that execute "concurrently." Dundas has only a single thread. Ukai's multiple user processes do not include one that is dedicated to "software prefetching" while others perform normal processing. Even if one could consider such user processes analogous to processor threads, all of them execute only what Applicant terms "normal threads" in the claim.

Independent claim 16 is amended to include the "run ahead mode separate from a normal processing mode." The identifiers then protect against eviction "during the run ahead mode." Ukai has no such modes. His complex OS-level file-buffering system can be analogized to Applicant's processor cache only in the light of Applicant's disclosure. But, even so, Ukai prefetches files and executes user processes concurrently, in clear contradistinction to the language of claim 16. Although Dundas prefetches and executes normally at different times, his system has no operation performed "during the run ahead mode." Dundas neither teaches nor requires such an operation, because he has no "identifiers" nor any other protection against overzealous prefetching during run-ahead processing. Indeed, Dundas recognizes no need for protection, and Ukai has neither suggests nor has any purpose for Dundas' separate run-ahead processing, and given the wide differences in implementation between their systems, one skilled in the art would not be motivated not be motivated to combine Dundas and Ukai except in light of Applicant's disclosure. The rejection under 35 USC §103 must fall for both of these reasons.

Claims 17-20 depend from claim 16, and incorporate all of its features, and distinguish the art for other reasons as well. For example, claim 18 specifies that at least one of the identifiers operates upon a cache line at two different times: at allocation "during the separate run ahead mode" and at use ("touched") "during the normal execution mode."

Applicant respectfully traverses the rejection of claims 21-24. Independent claim 21 recites "a plurality" of processors having "a plurality" of caches. Neither Dundas nor Ukai suggests more than one processor, and Ukai's operations occur within a single cache or buffer area. Neither reference suggests a "communication device" coupling multiple processors to a main memory. This memory is recited separately from the caches, and specifies that the protection bits are associated with the "caches." Ukai's protection, however, occurs in an area 10 "which OS 4 manages" (col. 4:8-9) in "the main memory" (col4:66-5:1). This difference is significant in that Ukai's method would be wholly impractical in a processor cache, and would not be considered for such usage by one skilled in the art. Claims 22-24 depend from claim 21.

Independent claim 25 is amended to recite the two modes explicitly. The processor has a run-ahead mode "separated in time" from the normal execution mode, whereas Ukai prefetches files and executes user processes concurrently, intermixed with each other in time. Claim 21 goes on to recite that the protection bit interdicts eviction "during operation in the run ahead mode." Dundas, of course, suggests no cache-line protection of any kind.

Claims 26-28 depend from claim 25, and recite additional features not found in Ukai. For example, claim 27 declares that the cache is located "on the same chip die as the processor." No one skilled in the art would consider Ukai's complex OS-level system for implementation in a fast chip-level cache; to extract only certain concepts from Ukai's system and then to modify them for Applicant's different environment would not occur without access to Applicant's disclosure.

New claims 29-31 recite a computer-readable storage medium for instructions to carry out methods according to the invention. The specification shows such a medium at 103 (Fig. 1), 616 (Figs. 6A-6C), and 704 (Figs. 7-8). The method elements of claims 29-31 distinguish the cited references for the same reasons as noted above in connection with claims 1-3, whence they are drawn.

Conclusion

For the foregoing reasons, Applicant urges that the Application meets all the statutory requirements, and respectfully requests reexamination and allowance. The Examiner is invited to telephone Applicant's attorney at(612) 373-6971 if this would facilitate prosecution of the Application.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 19-0743.

Respectfully submitted,

CHRISTOPHER B. WILKERSON

By his Representatives,

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. Box 2938
Minneapolis, Minnesota 55402
(612) 373-6971

Date 6 JAN 2004

By Michael
J. Michael Anglin
Reg. No. 24,916

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on this 6th day of January.

Anne M. Rich

Name: Anne M. Richards

Signature